

SISTEM KEAMANAN INSTALLER SOFTWARE DENGAN ACCESS CARD MELALUI LPT-1

Suryo Hadi ¹⁾

¹⁾ Universitas Kartini Surabaya, E-mail : Suryohadi_UKS@Plasa.Com

Abstract, *Advancements in hardware and software electronic technology bring such enormous effects that it is needed to add some protection system in designing an electronic device, especially in the software part. The protection system applied in this application uses an access card with microcontroller and interfacing through Line Port Terminal (LPT-1). With the use of this protection system, software distributions are expected to be safe from piracy.*

Keywords: *Protection, Software, Microcontroller*

Dewasa ini pada era globalisasi perkembangan ilmu pengetahuan dan teknologi berkembang dengan cepat dan pesat sehingga jarak dan waktu bukanlah menjadi kendala yang begitu menghambat. Dengan adanya perkembangan teknologi yang ada diusahakan dapat membawa hal yang positif bagi kehidupan manusia di dunia ini dan juga dengan berkembangnya ilmu pengetahuan maka informasi dapatlah tersebar dengan cepat dan efisien. Perkembangan di bidang elektronika khususnya di bidang mikrokomputer merupakan bidang yang sangat cepat sekali perkembangannya. Karena sudah menjadi sifat manusia untuk mencari jalan guna membantu mempermudah dalam kehidupan sehari-hari. Sebagai contoh dalam rumah tangga sudah banyak sekali peranan alat-alat elektronik yang berguna untuk membantu pekerjaan seperti mesin cuci, setrika listrik, VCD player, dan sebagainya. Dalam bidang industri peran alat-alat elektronik juga mengambil peran yang aling utama karena disamping untuk efisiensi kerja juga untuk meningkat mutu dan kualitas produksi. Misalnya untuk menentukan suatu hasil produksi yang akurat dan presisi dimana hal tersebut sulit sekali dilakukan oleh manusia maka alat-alat elektronik-lah yang menjadi solusinya.

Pada teknologi elektronik saat ini yang sudah menjadi *trend* adalah alat-alat yang sudah menggunakan sistem serba *digital*, karena disamping lebih mudah dalam pengoperasian alat, dimensi serta ukurannya dapat dirancang sekecil mungkin. Sebagai contoh jika kita melihat pada perkembangan teknologi telepon genggam (*Handphone*), semakin kecil dimensinya justru semakin cerdas. Sistem digital sendiri saat ini sudah banyak sekali yang

berbasis *Microprocessor* dan *Microcontroller* karena dari segi perancangannya sistem ini dikenal lebih mudah untuk dirancang dan mampu diprogram sendiri oleh si perancang. Pada penelitian ini penulis ingin melakukan suatu proteksi software dengan menggunakan access card yang dapat mencegah agar software yang akan kita distribusikan tidak dapat dibajak oleh pihak luar. Disamping itu hardware yang digunakan akan menggunakan IC mikrokontroler AT-89C51 yang memiliki dimensi kecil (20 pin) ini dapat diprogram secara perangkat lunak sehingga card proteksi ini juga aman dari pembajakan.

Pemilihan sarana komunikasi untuk mengakses card proteksi tersebut pada penelitian ini akan menggunakan jalur komunikasi paralel LPT1 yang biasanya digunakan untuk printer. Sehingga pemakaian software akan lebih umum baik pada komputer dekstop maupun komputer laptop.

PARAREL PORT

Kanal Pararel Port adalah kanal yang telah tersedia pada kumputer dan yang banyak sekali dipakai. Port ini sanggup melakukan pentransferan data hampir 9-bit atau 12-bit pada satu saat yang sama, sehingga hanya membutuhkan rangkaian eksternal yang minim untuk mengimplementasikan ke berbagai aplikasi. Komposisi dari port terdiri atas 4 buah jalur kontrol, 5 buah jalur status dan 8 buah jalur data. Port ini umumnya banyak dijumpai pada bagian belakang komputer menggunakan konektor jenis DB-25 betina.

Pararel port terakhir telah distandarisasi oleh IEEE 1284 sebagai standar pertama

pada tahun 1994. Standar ini mendefinisikan 5 modus pengoperasian yang diantaranya adalah sebagai berikut:

1. Modus Kompabilitas
2. Modus Nibble (4-bit)
3. Modus Byte (8-bit)
4. Modus ECP (*Extended Capabilities Port*)
5. Modus EPP (*Enhanced Parallel Port*)

Tujuannya adalah untuk merancang driver baru dan perangkat kompatibel dengan yang lain dan juga kompatibel dengan *Standard Parallel Port* (SPP). Mode Kompabilitas, Nibble dan Byte digunakan hanya pada perangkat standar yang tersedia pada *adapter* paralel port bawaannya, sedangkan EPP dan ECP mode membutuhkan tambahan perangkat keras yang dapat berjalan dengan kecepatan yang lebih cepat, namun tetap cocok dengan mode SPP. Modus kompabilitas hanya dapat mengirim data satu arah dengan kecepatan sekitar 50 Kbyte per detik tetapi dapat juga mencapai di atas 150 Kbyte per detik. Pada penerimaan data, kita harus mengubah dalam mode Nibble atau Byte. Nibble dapat menerima input 4-bit pada arah terbalik. Mode Byte menggunakan fitur paralel dua arah untuk meng-input-kan 1 byte data pada arah terbalik. Pengembangan dan perluasan Paralel Port menggunakan tambahan perangkat keras untuk menghasilkan dan mengatur protokol *handshaking*. Guna mengeluarkan 1 byte ke printer harus disesuaikan mode-nya dalam perangkat lunak. Hal ini memberikan batas kecepatan tertentu pada port yang sedang berlangsung. Adapun prosedur *handshaking* yang harus dilakukan adalah sebagai berikut:

1. Kirim data 1 byte ke Data Port.
2. Periksa apakah printer "busy", jika printer masih sibuk, maka tidak akan menerima data apapun.
3. Kirim sinyal "strobe" low. Hal ini memerintahkan printer bahwa ada data baru buatnya.
4. Membuat strobe high kembali setelah menunggu selama sekitar 5 mikro detik setelah "strobe" di-low-kan.

MIKROKONTROLER AT89C51

AT 89C51 adalah keluarga mikrokontroler 8bit, MCS 51, chip CMOS yang mengkonsumsi daya relatif kecil ini memiliki fasilitas yang jauh lebih baik dibandingkan dengan controller

dari keluarga MCS 51 lainnya. Chip ini mampu dioperasikan dengan frekwensi 0 MHz sampai 24 MHz. Rangkaian osilator dan pembangkit Clock terletak didalam chip ini. Mikrokontroler AT 89C51 merupakan pusat pengontrolan dari seluruh system yang ada. Keping pintar ini memiliki spesifikasi sebagai berikut:

- (a) Memiliki 2 Kilo byte Flash EEPROM, 1000 kali pemrograman.
- (b) Memiliki 128 byte RAM
- (c) Memiliki 24 bit programmable port I/O
- (d) Dua buah timers 16 bit
- (e) Memiliki serial interface
- (f) 5 buah sumber interrupt

Pin-pin IC AT 89C51, sebanyak 32 pin dari 40 pin yang dimiliki oleh AT 89C51 difungsikan sebagai jalur I/O. tetapi 24 jalur mempunyai 2 fungsi. Masing-masing dapat dioperasikan sebagai port I/O atau jalur kontrol atau dipakai sebagai bus alamat dan bus data dalam pengaksesan memori eksternal. masing-masing pin-pinnya mempunyai fungsi dan yang berbeda disesuaikan dengan rancangan yang kita buat. Adapun fungsi masing-masing pin-pin IC AT 89C52 adalah sebagai berikut :

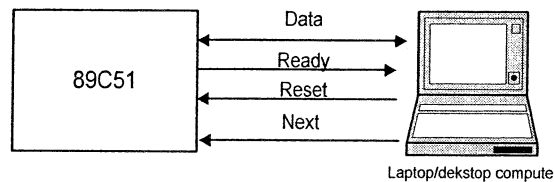
- (a) Port 0, merupakan port dua fungsi yang berada pada pin 32-39 dari IC AT 89C52. didalam rancangan sistim sederhana port ini digunakan sebagai port I/O serbaguna. Untuk rancangan yang lebih komplek dengan melibatkan memori eksternal jalur ini dimultipleks untuk bus data dan bus alamat.
- (b) Port 1, port ini disediakan sebagai port I/O dan berada pada pin 1-8.
- (c) Port 2, (pin 21-28) merupakan port dua fungsi yaitu I/O serbaguna atau sebagai bus alamat byte tinggi untuk rancangan yang melibatkan memori eksternal.
- (d) Port 3, port ini merupakan port dua fungsi yang berada pada pin 10-17, port ini mempunyai multifungsi
- (e) PSEN (*program store Enable*), adalah sebuah sinyal keluaran yang terdapat pada pin 29. fungsinya adalah sebagai sinyal kontrol untuk memungkinkan mikrokontroler membaca program (code) dari memori eksternal Biasanya pin ini dihubungkan ke pin OE EPROM. Jika eksekusi program dari ROM internal 8051/8052 atau dari flash memori AT 89C52, maka PSEN berada pada kondisi tidak aktif(high).

- (f) **ALE (Address Latch Enable)**, adalah sinyal output ALE yang berada pada pin 30 fungsinya sama dengan pada mikroprosesor INTEL 8085, 8088, 8086. sinyal digunakan untuk demultiplex bus alamat dan bus data. Sinyal ALE membangkitkan pulsa sebesar 1/6 frekwensi osilator dan dapat dipakai sebagai clock yang dapat digunakan secara umum. Jika clock AT 89C52 menggunakan crystal 12MHz maka sinyal osilator ALE sebesar 2MHz.
- (g) **EA (Eksternal Acces)**, masukan sinyal EA terdapat pada pin 31 yang dapat diberikan logic tinggi (+5V). jika EA diberikan logic tinggi maka mikrokontroller akan mengakses program dari ROM internal (Eprom / Flash memori). Jika EA diberikan logic rendah maka mikrokontroller akan mengakses program dari memori eksternal. Untuk IC mikrokontroller 8031/8032 EA harus diberikan logic rendah karena chip ini tidak memiliki ROM internal. Jika pin EA pada mikrokontroller 8051/8052 atau AT 89C52 diberi logic rendah maka ROM internalnya tidak berfungsi dan eksekusi program diambil dari memori eksternal. Pin EA pada 8051 atau AT 89C52 juga dipakai sebagai tegangan pemrograman EPROM atau Flash memori.
- (h) **RST (reset)**, adalah input reset pada pin 9 adalah reset master untuk 8051 atau AT 89C52. pulsa transisi dari tinggi ke rendah selama 2 siklus akan mereset mikrokontroller.
- (i) **Oscillator**, masukan yang disediakan pada chip dikemudikan dengan kapasitor crystal yang dihubungkan pada pin 18 dan 19. diperlukan kapasitor 30pF sebagai penstabil. besar nilai kapasitor ini sebesar 12 MHz - 24 MHz untuk AT89C52.
- (j) **Power**, AT 89C52/8051 dioperasikan pada tegangan suplay +5V, pin Vcc berada pada nomor 40 dan pin 20 untuk GND.

PERANCANGAN SISTEM

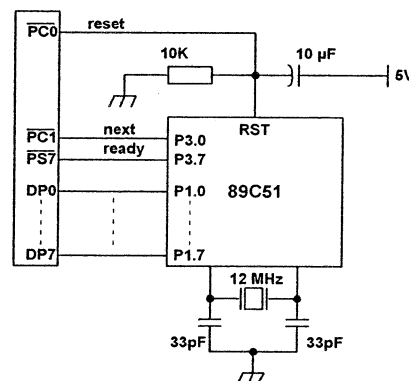
Pada gambar 1. diperlihatkan blok diagram perangkat keras keseluruhan sistem, yang dirancang, terdiri dari sebuah IC mikrokontroller 89C51 yang dihubungkan secara paralel dengan konektor LPT-1 pada komputer. Mikrokontroller sebelumnya telah diprogram agar dapat mengeluarkan urutan kode biner 8-bit secara paralel pada port LPT1.

Kode-kode biner tersebut nantinya akan dikonversikan menjadi dalam bentuk karakter ASCII yang akan membentuk data *String* sebagai data dari nomor seri card serta pasword yang digunakan saat software akan di-install. Pada penggunaannya card ini akan memiliki sebuah conector DB-25 yang dihubungkan langsung dengan conector DB-25 LPT1.



Gambar 1. Blok Diagram Alat

Mula-mula ketika *card* belum dihubungkan dengan LPT-1 maka kondisi dari Port Status (PS7) akan bernilai '0' hal ini akan menandakan pada komputer bahwa card tidak terpasang, sehingga sistem dapat mengabaikan atau membatalkan proses peng-install-an. Namun jika card telah dihubungkan dengan LPT-1 dan setelah mikrokontroller diberi catu daya dari sebuah *power supply* maka kondisi dari PS7 akan menjadi '1' seperti yang terlihat pada gambar skematik di bawah ini.



Gambar 2. Rangkaian Skematik Alat

Jalur data 8-bit (P1.0 – P1.7) dihubungkan sebagai input ke jalur Data Port LPT-1 (DP0 – DP7) dimana setiap *byte* berisi sebuah karakter. Setiap karakter dapat berubah-ubah sesuai urutan yang telah ditentukan oleh perangkat lunak (assembly) pada ROM internal 89C51. Urutan karakter-karakter tersebut akan berubah

dengan jalan memberikan pulsa dari '0' ke '1' melalui input P3.0 (next), dimana hal ini dilakukan menggunakan sebuah output dari LPT-1 yaitu (PC1). Sedangkan PC0 digunakan sebagai sinyal reset agar data yang keluar adalah yang terindikasi sebagai karakter pertama, kemudian PC1 memicu ke karakter-karakter berikutnya.

Dalam penelitian ini penulis membuat sebanyak tiga *card* dengan nomer seri yang berbeda masing-masing adalah "abcde-1234", "fghij-5678", dan "klmno-0912". Masing-masing nomer seri ini akan diprogram ke dalam ROM internal IC 89C51, untuk selanjutnya dapat diakses atau dibaca langsung melalui sistem perangkat lunak dari *installer* yang akan di distribusikan. Adapun program assembly 89C51 dari ketiga nomer seri di atas adalah sebagai berikut.

```
[1]      ;DONGLE_1.ASM
[2]      ;S/N: abcde-1234
[3]      ORG 00H
[4]      JMP MULAI
[5]      DELAY:  MOV R0,#0FFH
[6]      DEL:    DJNZ R0,DEL
[7]              RET
[8]      NEXT_HI: NOP
[9]              JNB P3.0,NEXT_HI
[10]             RET
```

Dst.

Pembuatan Paket Installer

Installer yang dibuat pada penelitian ini adalah installer dari sebuah program Media Player sederhana yang juga dibuat dengan Borland Delphi 5. Satu paket telah berisi sebuah file aplikasi MEDIAX.EXE (319 KB), File WAV sebanyak 108 file (2,05 MB), sebuah file MIDI (24 KB), dan dua file AVI (17,2 MB). Sehingga total kapasitasnya adalah sekitar 19,6 MB.

Untuk membuat sebuah paket installer pada penelitian ini digunakan *utility* INSTALL SHIELD for DELPHI. Adapun langkah-langkah pembuatannya adalah sebagai berikut:

1. Setelah menjalankan INSTALL SHIELD dan membuat project baru, kemudian kita dapat menentukan nama project beserta direktorinya.
2. Setelah muncul Setup Checklist, kita dapat mengatur fitur-fitur paket installer yang akan kita buat. Pertama adalah dengan meng-klik option **Application Information**, maka akan muncul kotak dialog **Set Visual Design** dimana terdapat tiga Tabs yaitu **App Info**, **Main Windows**, dan **Features**.

3. Men-set pada Tab App Info dengan Application Name: MediaX, Version:1.0, Company xxxx, Default Destination Directory: C:\MediaX. Kemudian kita dapat menentukan Application Executable dengan jalan meng-klik button Browse kemudian tentukan file MediaX.EXE yang akan dimasukkan.
4. Klik Tab Main Window untuk menampilkan tampilan awal dari installer MediaX. Kemudian set Main Title pada Text lalu isikan title yang akan ditampilkan. Set juga Background Color dengan 'Dithered Red'.
5. Klik OK untuk mengakhiri pengaturan Set Visual Design.
6. Ketika kembali pada tampilan, klik **Groups And Files**. Maka akan muncul kotak dialog **Specify Components And Files**. Lalu tambahkan 3 buah group dengan meng-klik button ADD GROUP dengan menambahkan group AVI, MIDI, dan WAVE. Masing-masing di-set pada Destination Directory yang sama yaitu [INSTALLDIR]\AVI, [INSTALLDIR]\MIDI, dan [INSTALLDIR]\WAVE. Setelah itu klik button Launch Explorer lalu *drag* file-file yang akan diikuti sertakan ke masing-masing group yang telah tersedia.
7. Klik OK.
8. Klik Option **Disk Builder** maka akan muncul kotak dialog. Karena installer yang dibuat nantinya akan disalin ke dalam CDROM maka kita juga harus men-set Disk Size: CDROM, kemudian memulai proses pembuatan installer dengan jalan meng-klik button Build. Proses ini memerlukan waktu beberapa menit karena selama waktu tersebut Installshield akan melakukan pemampatan (*Compressing*) dari seluruh file penunjang yang disertakan. Hasil dari pembuatan paket installer ini dapat kita coba dengan menjalankan SETUP.EXE yang diletakkan pada directory C:\My Documents\Installer MediaX\MediaX\650MB\Disk1.

Akses Port Menggunakan Borland Delphi

Seperti telah dijelaskan sebelumnya bahwa pengaksesan card proteksi (89C51) secara perangkat keras adalah harus sesuai dengan timing diagram. Pada pembahasan berikutnya akan dibahas mengenai pengaksesannya secara

perangkat lunak. Perangkat lunak (software) untuk mengoperasikan alat ini ini ditulis dan dikompilasi menggunakan Borland Delphi 5.0 yang merupakan pengembangan dari bahasa Pascal. Guna mengoperasikan transfer data menggunakan address port printer (LPT-1) maka diperlukan penggabungan bahasa assembly dalam program yang dibuat karena Delphi sendiri tidak mempunyai perintah yang dapat digunakan sebagai akses port.

Berikut ini adalah sebuah sub-rutin (prosedur) program yang menggunakan bahasa assembly dalam program utama.

```
[1] Procedure OutPort(AddrOUT:word;DataOut:byte);
[2] Begin
[3]   Assembly
[4]     MOV DX,AddrOut {Register DX <- AddrOUT}
[5]     MOV AL,DataOUT {Register AL <- DataOut}
[6]     OUT DX,AL
[7]   end;
[8] end;
[9] Function InPort(AddrIN:word):byte;
[10] Var Kode:byte;
[11] Begin
[12]   Assembly
[13]     MOV DX,AddrIN {Register DX <- AddrIN}
[14]     IN AL,DX {Register AL <- Port [DX]}
[15]     MOV Kode,AL
[16]   end;
[17] InPort:=Kode;
[18] end;
```

Pada baris 1 - 8 adalah prosedur untuk mengirim data melalui output port. Prosedur OutPort ini masih memiliki dua parameter yaitu yang pertama (AddrOUT), yang bertipe word, sebagai pencatat address port 0000H-FFFFH sedangkan parameter yang kedua (DataOut) sebagai data yang akan dikirim melalui port yang bersangkutan. Misalnya prosedur ini ingin digunakan untuk mengirim data AAH ke Data Port LPT-1, maka statement yang diberikan adalah : **OutPort(\$378,\$AA)**.

Statement tersebut akan dikerjakan oleh prosedur OutPort pada baris 4 sampai 6. Address 378H (penulisan bilangan hexa dalam Delphi "\$378") akan dimasukkan pada register DX, yaitu register internal prosesor yang berfungsi sebagai pencatat address port I/O. Kemudian data E0H akan dimasukkan pada register akumulator AL, yaitu register yang berfungsi sebagai lintas data ke port. Perintah OUT pada baris 6 mengakibatkan data yang tersimpan dalam register AL akan dikirim ke address port yang alamatnya ditunjuk oleh isi register DX.

Pada baris 9 sampai 18 merupakan fungsi untuk membaca data dari perangkat luar melalui port input. Fungsi (*Function*) dalam bahasa pascal sedikit berbeda dengan Procedure, dimana fungsi memiliki kembalian data sedangkan Procedure tidak memiliki kembalian data. Kembalian data ini nantinya akan ditampung dalam sebuah *variable* sehingga pemakaian statement-nya juga berbeda dengan procedure. Misalkan data dari luar akan dibaca melalui Port Status LPT-1 dan akan ditampung pada variable "DataTerbaca", maka statement yang diberikan adalah :

DataTerbaca:=InPort(\$379).

Statement tersebut akan dikerjakan oleh fungsi InPort pada baris 13 sampai 17. Address 379H akan dimasukkan pada register DX kemudian pada perintah IN pada baris 14 mengakibatkan data akan tersimpan dalam register AL dari address port yang alamatnya ditunjuk oleh isi register DX.

Pernyataan pada baris 15 berfungsi untuk menampung isi AL (hasil pembacaan) ke variable "Kode" untuk selanjutnya di baris 17 isi Kode ditransfer sebagai kembalian data-nya.

Siklus Kerja Pengaksesan Port

Sesuai dengan timing kerja, maka pada perangkat lunak harus terlebih dahulu diberi suatu tetapan sebagai sinyal kerja dari setiap port. Pemberian data pada PC0 sampai dengan PC3 haruslah disesuaikan pula dengan sinyal pengatur arah lintasan data pada PC5, dalam hal ini Data Port akan digunakan sebagai port input maka PC5 harus dibuat '1'. Adapun tabel pendataan dari setiap siklus adalah sebagai berikut:

Tabel 1. Tetapan Data Untuk Port PC

PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	Data	Akses
0	0	1	0	1	0	1	1	2BH	None, Mode Input
0	0	1	0	1	0	1	0	2AH	Reset
0	0	1	0	1	0	0	1	29H	Next

Selain itu kita juga harus menetapkan masukan PS-7 dimana data dari Port Status ini tetap dibaca menggunakan modus 8-bit, sehingga untuk melakukan penyaringan diperlukan metode *bit masking*. Hal ini bertujuan agar sistem hanya mengetahui kondisi logika dari sebuah bit saja sedangkan ke 7-bit yang lainnya di abaikan.

Bit Masking ini sendiri pada sistem perangkat lunak difungsikan sebagai suatu

prosedur yang di akses pada tampilan awal yaitu *Form On Active* seperti pada sub rutin di bawah ini.

```
[1] procedure      TSetup_X.FormActivate(Sender:
TObject);
[2]   var karakter: array [1..10] of char;
[3]       KarKe:byte;
[4]   begin
[5]       If (InPort(PortStatus) and $80) = $80 then
[6]           Begin
[7]               ResetDongle;
[8]               SerialNum:=0;
[9]               For KarKe:=1 to 10 do
[10]                  begin
[11]                      Karakter[KarKe]:=GetChar;
[12]                      SerialNum:=SerialNum
Karakter[KarKe];
[13]                  end;
[14]              end else
[15]                  Begin
[16]                      ShowMessage('No Card!!');
[17]                      Application.Terminate;
[18]                  end;
[19]      end;
```

Pada Statement IF-THEN di baris ke 5 nampak bahwa isi dari Port Status ini dibaca kemudian di-AND-kan dengan 80H apabila hasil dari operasi ini bernilai 80H maka program akan membaca sebanyak 10 karakter dari Card (baris 9 sampai 13) dan menggabungkannya kedalam variable bertipe *string* yaitu *SerialNum*. Sebaliknya jika hasil operasi bukan bernilai 80H maka sistem akan menganggap Card tidak ada kemudian di baris ke 17 proses dibatalkan dengan menggunakan statement *Application.Terminate*.

Pengoperasian File dan Direktori

Paket Installer yang dibuat sebenarnya memiliki 2 buah aplikasi. Aplikasi pertama berfungsi sebagai installer yang telah dibuat sebelumnya menggunakan InstallShield. Dan aplikasi yang kedua adalah yang berfungsi sebagai pengakses Card serta sebagai aplikasi yang memanggil file installer SETUP.EXE. Akan tetapi untuk kasus ini file installer SETUP.EXE harus diganti nama file dan ekstensinya agar pemakai tidak tahu bagaimana meng-install secara langsung.

Semula file SETUP.EXE yang telah diubah nama dan ekstensinya disimpan ke dalam CD, kemudian **SETUPX.EXE** buatan dari Delphi dijalankan akan membuat direktori sementara dalam hardisk lalu mengkopi semua file-file penunjang yang ada. Setelah itu installer dijalankan sampai selesai, jika peng-installan

berhasil barulah direktori sementara beserta isinya dihapus.

Dalam aplikasi ini diperlukan adanya beberapa fungsi program dari unit *FileCtrl* yang dapat dipakai antara lain:

- **FileExists(const FileName: string): Boolean**, gunanya untuk mengecek apakah suatu file ada dalam suatu direktori. Nilai baliknya bernilai *true* apabila file ditemukan, dan bernilai *false* jika tidak ditemukan.
- **DirectoryExists(const Name: string): Boolean**, gunanya untuk mengecek apakah suatu direktori ada. Nilai baliknya bernilai *true* apabila direktori ditemukan, dan bernilai *false* jika tidak ditemukan.
- **CopyFile()**, gunanya untuk mengkopi file ke direktori tujuan. Fungsi ini memiliki tiga buah parameter yaitu:
 - **lpExistingFileName: Pchar**, sebagai penunjuk file yang ingin dikopi.
 - **lpExistingFileName: Pchar**, diisi dengan direktori tujuan.
- **BFailIfExists: Boolean**, menunjuk aksi yang akan diambil jika file dengan nama yang sama telah ada pada lokasi tujuan. Jika diisi dengan *false* maka file yang sudah ada akan ditimpa (*overwrite*), jika *true* file yang ada tidak ditimpa.
- **DeleteFile(const FileName: string): Boolean**, gunanya untuk menghapus file yang berada dalam suatu direktori.
- **RenameFile(varOldFileName,NewFileName: string)**, gunanya untuk mengganti nama file dan ekstensinya.
 - **CreateDir(const Dir: string)**, gunanya untuk membuat direktori baru.
 - **RemoveDir(const Dir: string)**, gunanya untuk menghapus direktori.

Algoritma Program Utama

- Inisialisasi LPT-1 dimana Data Port digunakan sebagai input 8-bit, dengan jalan men-set PC5 menjadi '1'.
- Check keberadaan card melalui PS7, jika bernilai '1' maka berarti card terpasang maka lanjutkan ke langkah 3. Namun jika PS7 bernilai '0' berarti card tidak terpasang, maka proses dihentikan.
- Reset Card, untuk mengambil karakter pertama. Memberikan pulsa tinggi ke PC0.

- Melakukan pengambilan karakter berikutnya dengan memberikan pulsa tinggi ke pada PC1.
- Pengambilan karakter pada langkah 4 di ulangi sampai 10 kali, kemudian kesepuluh karakter tersebut digabung menjadi sebuah serial number.
- Memasukkan input serial number. Jika serial number salah maka proses dihentikan. Apabila serial number benar jalankan Installer.
- Selesai.

PENGUJIAN & PENGOPERASIAN PROGRAM

Pengujian

Sebelum mengakses perlu diketahui bahwa kita harus memastikan setting BIOS pada komputer untuk mode port rinter adalah EPP. Setelah itu kita dapat menjalankan aplikasi SETUPX.EXE kemudian menekan tombol Ctr-Alt-T maka akan muncul tampilan. Agar kita dapat juga mengamati secara langsung kita juga dapat menggubungkan setiap bit pada jalur Data Port dengan 8 buah indikator LED. Kedelapan indikator ini akan menampilkan nilai setiap bit. Jika LED menyala maka nilainya adalah '1' dan jika padam maka nilainya adalah '0'

Sebagai contoh jika card yang dipakai mempunyai nomer seri 'abcde-1234' maka secara berurutan keluaran dari card adalah '0110 0001', '0110 0010', '0110 0011', '0110 0100', '0110 0101', '0010 1101', '0011 0001', '0011 0010', '0011 0011', '0011 0100'.

Pengoperasian Installer

Pada tampilan awal saat SETUPX.EXE dijalankan disini kita dapat langsung memasukkan serial number yang sesuai dengan card bawaannya. Setelah serial number dimasukkan lalu klik button *Continue*.

Jika serial number yang dimasukkan benar dan sesuai dengan nomor seri card maka sistem akan menjalankan aplikasi installer. Namun sebelumnya aplikasi tersebut telah dimasukkan dalam hardisk pada direktori "c:\a_endah\" yang mana di dalamnya berisi file-file penunjang beserta file SETUP.EXE. Semula file-file tersebut dikopi dari CD. Sementara MediaX Setup dijalankan aplikasi pemanggil akan menunggu sampai proses instalasi selesai dan untuk kemudian sistem akan menampilkan pesan bahwa proses instalasi telah berhasil, lalu terakhir akan membuat sebuah *Shortcut*. Kita

dapat menjalankan aplikasi MediaX yang telah di-install langsung dengan meng-klik dua kali *shortcut* MediaX.

SIMPULAN

Port paralel LPT-1 memiliki kelebihan yaitu kita dapat memanfaatkan jalur data dua arah (*bidirectional*) 8-bit dengan jalan menset bit Port Status PS7. Dengan menggunakan mikrokontroler kita dapat dengan mudah memprogram fungsi-fungsi yang kita kehendaki. Selain itu mikrokontroler juga memiliki kelebihan yaitu kita dapat melakukan proteksi terhadap program yang tersimpan di dalamnya agar tidak dapat diketahui oleh pihak lain.

DAFTAR RUJUKAN

- [1] Ayala, Kenneth J., **The MCS-51 Family**, West Publishing Co., Singapore, 1993.
- [2] Coughlin, Robert F., Frederick F. Driscoll, Herman W., **Penguat Operasional dan Rangkaian Terpadu Linear**, Erlangga, Jakarta, 1992.
- [3] Djoko Pramono, **Mudah menguasai Delphi 3.0.**, Elex Media Komputindo, Jakarta, 1999.
- [4] Green, DC., **Data Communication**, Longman Group, United Kingdom.
- [5] Jogivanto H.M., **Teori dan Aplikasi Program Komputer Bahasa Turbo Pascal (jilid 1 dan 2)**, Andi Offset, Yogyakarta, 1995.
- [6] KF Ibrahim, **Basic Electronic System**, Pitman Publishing Ltd., United K., 1982
- [7] Lister, Paul, **Microprocessor Fundamentals**, Pitman Publishing, Singapore, 1993.
- [8] Malik, Moh. Ibnu., **Bereksperimen Dengan mikrokontroler MC-51**, Elex Media Komputindo, Jakarta, 1997.
- [9] Mottershead, Allen, **Electronic Devices and Circuit**, Prentice Hall of India, New Delhi, 1990.
- [10] Santosa, P. Insap, **Grafika Komputer dan Antarmuka Grafis**, Andi Offset, Yogyakarta, 1996.
- [11] _____, **Interfacing the Standart Parallel Port**, <http://www.senet.com.au/~cpeacock>